# Conflict Resolution in Warehouse Robotics

Goutham Muralikrishnan, Noah Steinkrüger, Vinay Orsu

Videos and more information »

## Abstract

Mobile robots deployed in warehouses must navigate safely and efficiently in close proximity, sometimes without reliable global coordination.

We present a fully decentralized conflict detection and resolution module that runs onboard each robot. During nominal navigation, each robot monitors LiDAR scans odometry, commanded velocities. When a conflict is triggered, a lightweight quadratic program (QP) filters the nominal velocity using LiDAR-derived barrier constraints for collision avoidance and a liveness (yielding) constraint to break symmetric deadlocks.

We evaluate the approach in multi-robot simulation scenarios with static and dynamic obstacles with randomized initialization perturbations. Results show our approach with the detection and resolution performed better when compared to a baseline without detection-controlled arbitration.

## Problem Context

Existing multi-robot navigation follows two directions: **centralized coordination** can be globally consistent but requires shared state and scales poorly in dense traffic, while **decentralized reactive avoidance** is fast and local but can stall in narrow passages and symmetric encounters. **Hybrid** approaches switch to stronger coordination after deadlock, but add integration complexity and can cause runtime spikes. These gaps motivate a lightweight onboard module that combines early local detection with liveness-aware resolution, improving robustness across scenarios without changing the navigation stack. Figure 1 shows example conflict scenarios caused by robot–robot interactions or obstacles.

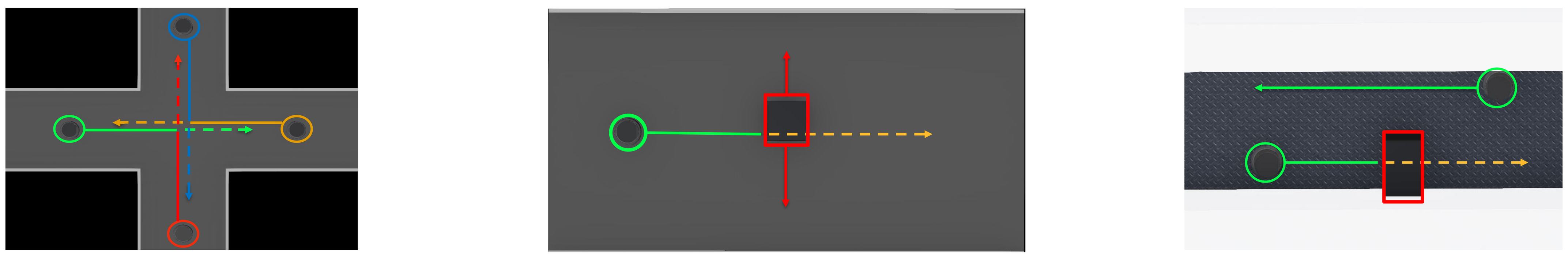

**Figure 1**: Example scenarios

## Methods

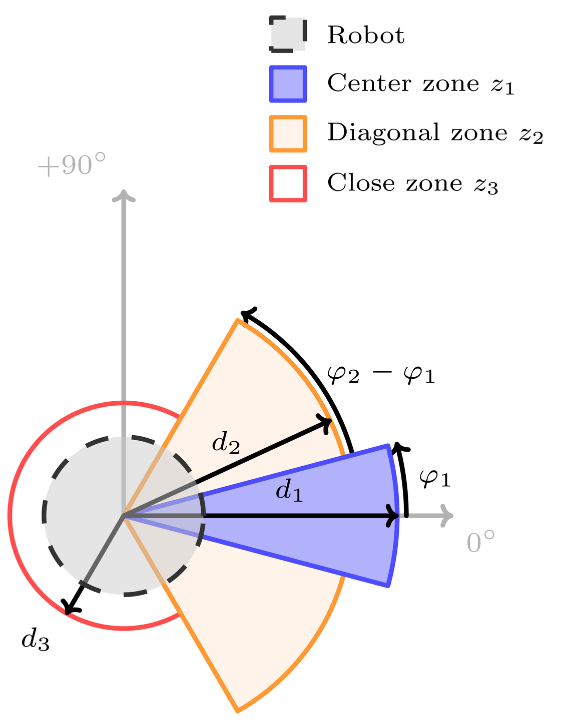


**Figure 2**: Detection zones

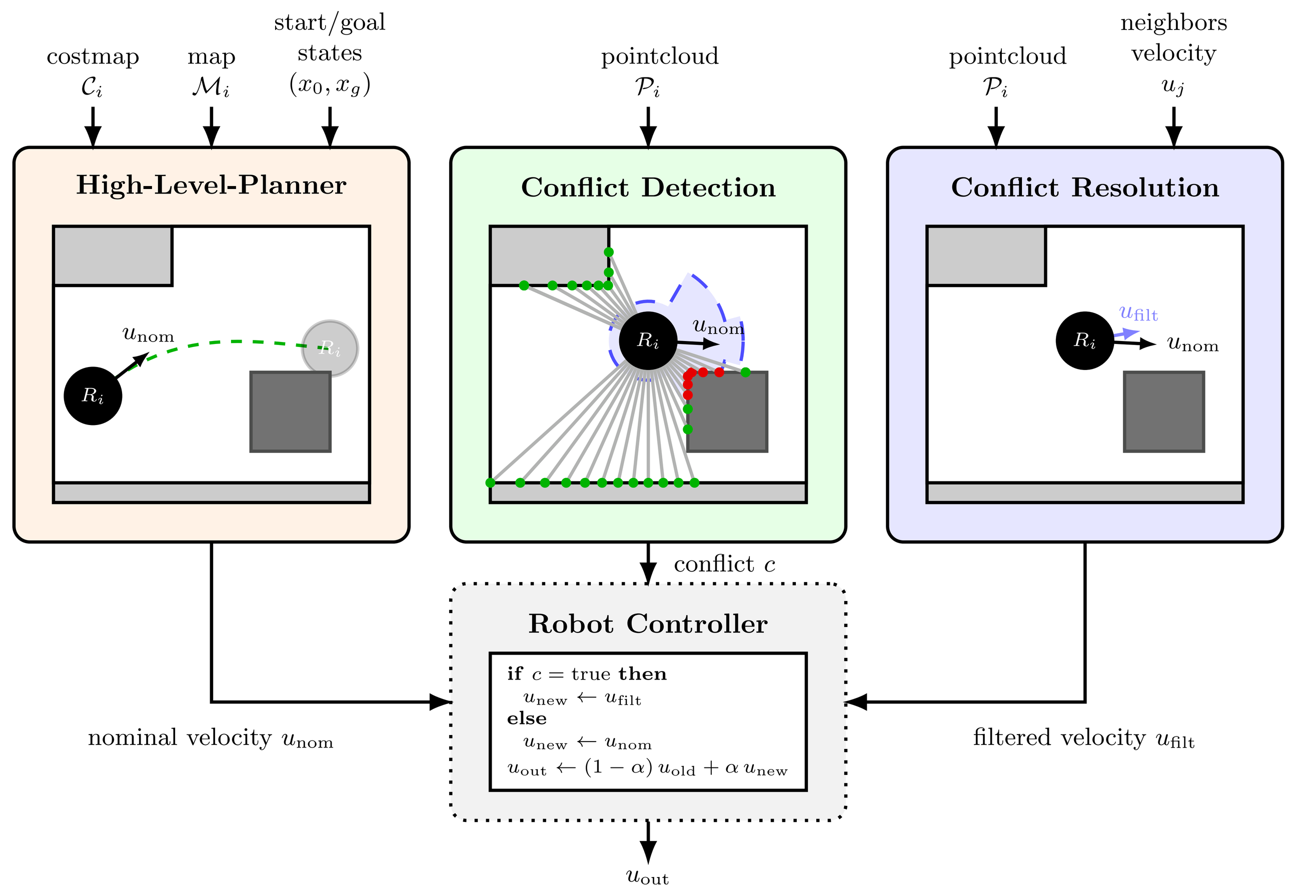


**Figure 3**: System architecture

### Detection

**Input**:

- Pointcloud $\mathcal{P}_i = \{(x_k, y_k)\}_{k=1}^{M}$

**Conflict Detection Region**:

- LiDAR safety-region detector with forward sectorization + close zone. (See Figure 2.)

**Region decomposition**:

$$z_1 = \{\phi : |\phi| \le \varphi_1\}$$
$$z_2 = \{\phi : \varphi_1 < |\phi| \le \varphi_2\}$$
$$z_3 = \{-\pi < \phi \le \pi\}$$

**Hit indicator**:

$$\mathrm{hit}_i(\phi) = \mathbb{1}(r(\phi) < d_i), \quad \phi \in z_i$$

A conflict detection flag $c$ is raised when the hit of any of the region is true, once the flag is raised the robot controller transitions the velocity to the filtered velocity from the resolution module and switches back when conflict is cleared.

### Resolution

Computes a new velocity $u = [v, \omega]$ that stays as close as possible to the planner command $u_{\mathrm{nom}}$ while ensuring safety and progress.

$$\min_{v,\omega} (v - v_{\mathrm{nom}})^2 + (\omega - \omega_{\mathrm{nom}})^2$$

Subject to:

- **LiDAR safety** (barrier constraint):
  $$h_k = x_k^2 + y_k^2 - \delta_{\mathrm{eff}}^2, \qquad \delta_{\mathrm{eff}} \in \mathbb{R}$$
  $$\dot{h}_k + \alpha h_k \ge 0, \qquad \alpha \in \mathbb{R}$$
  Keep safe distance from obstacles.
- **Deadlock yielding** (if detected):
  $$v \le \frac{1}{\zeta} \|v_j\|, \qquad \zeta \in \mathbb{R}$$
  where $j$ indicates the dominant neighbor. Deadlock detection via relative motion geometry and velocity stagnation.
- **Velocity bounds** of the robot.

The QP runs in real time onboard each robot and outputs the filtered command to the controller (see Figure 3).

## Experiments

| | Environment | # of robots | # of obstacles | Detection + Resolution | | Resolution | |
|---|---|---|---|---|---|---|---|
| | | | | SR (↑) | $T_{\mathrm{res}}$ [s] (↓) | SR (↑) | $T_{\mathrm{res}}$ [s] (↓) |
| | Corridor | 2 | 0 | 1.0 | 20.35 ± 15.34 | - | |
| | Crossing | 2 | 0 | 0.7 | 39.53 ± 29.50 | 0.5 | 0.49 ± 0.63 |
| | Crossing | 3 | 0 | 0.9 | 66.29 ± 55.16 | 0.1 | 6.13 ± 0.00 |
| | Crossing | 4 | 0 | 0.4 | 62.27 ± 41.47 | 0.0 | |
| static | Corridor | 1 | 1 | 1.0 | 17.27 ± 8.18 | 1.0 | 0.53 ± 0.69 |
| static | T-Junction | 3 | 1 | 0.8 | 52.13 ± 22.89 | - | - |
| static | T-Junction | 5 | 1 | 0.8 | 116.33 ± 31.20 | 0.0 | - |
| static | Warehouse | 2 | 1 | 0.3 | 97.55 ± 79.11 | 0.0 | - |
| static | Warehouse | 3 | 1 | 0.1 | 124.98 ± 0.0 | - | - |
| dyn | Corridor | 1 | 1 | 0.0 | - | 0.0 | - |
| dyn | Crossing | 3 | 1 | 0.1 | 135.93 ± 0.00 | - | - |

We tested the combined detection and QP-based resolution module in different scenarios with varying robot counts and static or dynamic obstacles, performing 10 runs per setup and measuring collision-free success rate (SR) and resolution time $T_{\mathrm{res}}$. Compared to a resolution-only baseline, our approach achieved higher success rates in multi-robot interactions, particularly in dense crossings where the baseline frequently stalled. Performance remained strong in structured environments with static obstacles but degraded in highly constrained or scenarios where dynamic obstacles are included.

## Conclusion

We introduced a fully decentralized, two-layer module where each robot detects conflicts from LiDAR and filters the nominal velocity with a lightweight QP, This keeps the navigation stack modular.

In the 10-run trials, adding detection-triggered switching made the resolution behavior much more reliable than running the resolution module alone. Practically, this means robots were better at knowing when to stop following the nominal planner and when to apply the conflict filter, which helped them break symmetric stand-offs (where both robots react the same way and neither yields) and recover from close interactions. The remaining drop in performance in the warehouse setting suggests the method is most stressed when the environment is tight and cluttered, where interactions can repeat and LiDAR visibility can be less consistent. The key takeaway is that the switching logic is important for liveness, while dense layouts still expose perception- and interaction-limits.

Future work will focus on systematic parameter tuning, evaluation in denser/dynamic environments, stronger handling of difficult multi-agent interactions, and real-robot validation.

## Acknowledgements

We sincerely thank our supervisor, **Dr. Tanja Kaiser**, for her guidance and support throughout this work.

We also acknowledge our company partner, **symovo GmbH**, for their collaboration and valuable contributions.

Special thanks to **Tobias Fink** and **Dirk Braunschweiger** for their exceptional support.

## References

- A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control Barrier Function-Based Quadratic Programs for Safety-Critical Systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.
- J. Chen and R. Chandra, "LivePoint: Fully Decentralized, Safe, Deadlock-Free Multi-Robot Control in Cluttered Environments with High-Dimensional Inputs," 2025.
- E. Ferrera, Á. R. Castaño, J. Capitán, P. J. Marrón, and A. Ollero, "Multi-Robot Operation System with Conflict Resolution," in *ROBOT2013: First Iberian Robotics Conference*, ser. Advances in Intelligent Systems and Computing, M. A. Armada, A. Sanfeliu, and M. Ferre, Eds. Cham: Springer, 2014, vol. 252, pp. 407–419.
- E. Ferrera, J. Capitán, Á. R. Castaño, and P. J. Marrón, "Decentralized Safe Conflict Resolution for Multiple Robots in Dense Scenarios," *Robotics and Autonomous Systems*, vol. 91, pp. 179–193, 2017.